Abstract
We present a pose estimation algorithm for 2D input images which is based on probabilistic 3D shape models. Our underlying model is a pose-invariant category model comprised of 3D contour fragments. These 3D contour fragments are represented by probability density functions, which are described by Gaussian Mixture Models. Our pose estimation algorithm consists of two steps. First, we use an Unscented Transformation to build 2D aspect models from the pose-invariant 3D category model. Second, we introduce a novel similarity measure between Gaussian Mixture Models which is based on a hypothesis test. We demonstrate our pose estimation approach on seventeen poses of the ETH80 database for the two categories ‘horse’ and ‘cow’.

1. Motivation and Introduction
Objects differ significantly when they are viewed from different poses. Therefore, pose-invariant representations of objects and object categories gain more and more importance for vision tasks as object recognition, categorization or pose estimation. Many of the current 2D based object categorization systems are view dependent and each view has to be learned independently. Especially shape-based categorization systems [5, 7, 13] are sensitive to pose/view changes because the visual rim changes significantly on the object (see Figure 1).

To overcome these known problems of 2D based systems, multi-view and 3D model based categorization and pose estimation systems have been introduced in [6, 14, 1, 12, 15, 8]. Liebelt and Schmid [6] present a multi-view object class detection system where they combine a 2D part-based categorization approach with a 3D geometry-based model built from synthetic data (CAD models) for pose estimation. Stark et al. [14] also present multi-view object categorization built on 3D CAD Models.

*This work was supported by the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.
Arie-Nachmison and Basri [1] build a 3D implicit shape model for pose estimation of cars. In contrast to our method, the reconstruction of the 3D model does not work fully automatically. Savarese and Fei Fei [12] developed an approach where they build a 3D part based model from images of different poses for categorization and pose estimation. They present also a synthesis of object categories. We want to point out, that all of the systems mentioned above are appearance based. In contrast, our approach for pose estimation on 2D images is purely shape-based in that sense that we use the visual rim of objects and inner contour fragments as shape information. Our algorithm is based on probabilistic 3D contour category models for the categories ‘horse’ and ‘cow’ [10] which use a 3D shape representation based on 1D manifolds in 3D [9], so called ‘3D contour clouds’. We use an Unscented Transformation to compute probabilistic 2D contour models per aspect from these 3D shape models. Based on these models, we compute a similarity measure and establish a voting procedure to identify the pose of a given input image. Our experimental evaluation uses the available ground truth of the horses and cows of the well-known ETH80 database. The only assumption we make is that the objects category and position is already known from a categorization done beforehand.

2. 3D Pose estimation algorithm

Our pose estimation algorithm consists of two steps. First, we build 2D aspect models from 3D Gaussian Mixture Contour Models using an Unscented Transformation, which approximates a probability density function which undergoes a nonlinear transformation using just a small number of weighted points. Therefore, we can estimate the mean and the variance of our Gaussian Mixture Models (GMMs) in our projected 2D aspect models. Second, we use a voting procedure on the basis of these probabilistic 2D aspect models to estimate the pose of an object of a given 2D input image.

2.1. 3D Gaussian Mixture Contour Model

3D probabilistic contour models as described in [10] build the basis for our pose estimation algorithm. There, 3D contour fragments are generated from stereo images sequences and build so-called ‘3D contour clouds’ [9]. These 3D contour fragments are represented by probability density functions using GMMs, so that

\[ \Theta_K = \sum_{k=1}^{K} \alpha_k N(\mu_k, \Sigma_k), \]  

where \( \Theta_K \) is the GMM for a contour fragment \( F_i \) of a 3D contour cloud \( CC \) and \( K \) is the number of mixture components. Each component is a Gaussian distribution \( N(\mu_k, \Sigma_k) \) with mean \( \mu_k \), covariance \( \Sigma_k \), and weight \( \alpha_k \). A 3D Gaussian Mixture contour category model consists of a discriminative partition of probability densities. In contrast to the Kullback-Leibler divergence which takes the whole GMMs into account, partitions of probability densities are used and a discrete similarity measure is defined on them. A partition is a subset of size \( M \) of mixture components and is given by

\[ \Theta_M = \sum_{m=1}^{M} \alpha_m N(\mu_m, \Sigma_m) \quad \Theta_M \subseteq \bigcup \Theta_K, \forall F_i \in CC. \]  

Such a partition is learned from a set of 3D Gaussian Mixture Contour Models of specific objects of a category using a random feature selection algorithm. The test statistics is based on a hypothesis test for the similarity measure between GMMs. The similarity measure between densities is based on relative pairwise relations between mixture components and their principal eigenvector orientation. Then, a 3D Gaussian Mixture contour category model consists of a set of partitions of size \( Q \)
\[ \Theta_{\text{Category}} = \{ \Theta^q_M, q = 1, \ldots, Q \}, \]  

where \( \Theta^q_M \) is a partition of size \( M \).

### 2.2. Unscented Transformation

For 3D to 2D projection of a 3D Gaussian Mixture contour category model we use an Unscented Transformation (see [3]). An Unscented Transformation computes the mean and the variance of a random variable which has been transformed by a transformation \( f \). Let \( X \) be an \( L \)-dimensional random variable with mean \( \mu_k \), and variance \( \Sigma_k \), so that \( y = f(X) \). In our case the transformation \( f \) is a perspective projection. For the Unscented Transformation, first we compute \( 2L + 1 \) weighted points, so called sigma points, using the following equations:

\[
\begin{align*}
\xi_0 &= \mu_X \\
\xi_i &= \mu_X + (\sqrt{(L + \kappa \Sigma_X)}_i \\
\xi_i &= \mu_X - (\sqrt{(L + \kappa \Sigma_X)}_i, \\
\tau_0 &= \frac{\kappa}{L+\kappa} \\
\tau_i &= \frac{1}{2(L+\kappa)} \\
\end{align*}
\]

where \( \tau_i \) is the weight of the \( i^{th} \) sigma point with \( \sum_{i=0}^{2L} \tau_i = 1 \) and \( \kappa \) is a scaling factor\(^1\). The notation \( (\sqrt{(L + \kappa \Sigma_X)}_i \) means that we choose the \( i^{th} \) row of the matrix for the computation of the sigma point. Now, we compute the mean and the variance of our projected sigma points \( Y_i = f(X_i) \) by

\[
\begin{align*}
\mu_y &= \sum_{i=0}^{2L} \tau_i Y_i \\
\Sigma_y &= \sum_{i=0}^{2L} \tau_i (Y_i - \mu_y)(Y_i - \mu_y)^T. \\
\end{align*}
\]

### 2.3. Pose estimation

Our pose estimation algorithm requires a similarity measure between partitions of Gaussian Mixture Models to identify the pose of an object in a given 2D input image. Our novel similarity measure constitutes a modification of the well known Kullback-Leibler (KL) divergence, which is used in many existing shape matching methods as the divergence between two GMMs. There exists no closed-form solution of the KL-divergence between two Gaussian Mixture Models, but an approximation is suggested in [2] by:

\[
KL(f \| g) \approx \sum_{i=1}^{n} \alpha_i \min_j (KL(f_i \| g_j) + \log \frac{\alpha_i}{\beta_j}),
\]

where \( f \) and \( g \) are two GMMs:

\[
\begin{align*}
f &= \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} \alpha_i N(\mu_i, \Sigma_i) \\
g &= \sum_{j=1}^{m} g_j = \sum_{j=1}^{m} \beta_j N(\mu_j, \Sigma_j). \\
\end{align*}
\]

The KL-divergence between two GMMs uses the closed-form Kullback Leibler divergence between two Gaussians \( N(\mu_1, \Sigma_1) \) and \( N(\mu_2, \Sigma_2) \). It is given by

\[
KL = \frac{1}{2} \left( \log \frac{|\Sigma_2|}{|\Sigma_1|} + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2) \right)
\]

where \( |.| \) denotes the determinant. For several reasons, the approximation of the KL-divergence is not directly applicable. First, missing contour fragments (consequently, missing mixture components,  

\(^1\)In our experiments \( L = 3 \) and we choose \( \kappa = 1 \)
e.g. due to self-occlusion) have a big influence on the divergence between two GMMs. Second, in the KL-divergence between two mixture components, the position of a density has more influence than the orientation of the principal eigenvector of the variance.

In the remaining section we use the following notation: $I$ denotes a given input image containing an object of a given category. $P = P_1, ..., P_C$ denotes the 2D aspect models of $C$ poses. $(\mu_j^{P_i}, \Sigma_j^{P_i})$ and $(\mu_i^I, \Sigma_i^I)$ are mixture components of 2D aspect model $P_j$ and an input image $I$. Further, we define $oc_i$ to be the object center of the detected object in input image $I$ and we define $oc_j$ to be the object center of the aspect pose $P_j$. Let $v_j^{P_i}$ be the relative difference between a mixture component $\mu_j^{P_i}$ and the object center $oc_j$, and $v_i^I$ be the position difference between a mixture component $\mu_i^I$ and the object center $oc_i$. $(R_i, T_i)$ denotes the local transformation i.e. rotation and translation of a single mixture component $(\mu_i, \Sigma_i)$.

For our pose estimation algorithm we introduce

1. $SM(f_i, g_j)$: Similarity measure between mixture components, where the orientation of a mixture component (given by the principal eigenvector of its variance), and hence the orientation of the contour fragment, as well as its position on the object have the same importance.

2. $SM(f \parallel g)$: Similarity measure $SM(f \parallel g)$ between two GMMs. It is defined on the basis of the KL-divergence approximation (6), but by using $SM(f_i, g_j)$ we introduce a discretization.

3. $H(P)$: Pose hypothesis $H(P)$. Here, we use $SM(f \parallel g)$ as similarity measure between the GMMs of contour fragments of a given input image $I$ and a 2D aspect model $P_j$. In contrast to the 3D contour category model we do not use the partitioning but the GMMs as a whole.

Similar to [10] we introduce a similarity measure $SM(f_i, g_j)$ between two Gaussians which takes into account the orientation of the eigenvector and the position of the mixture component on the object.

Due to a first alignment of the bounding boxes of an aspect model and the object in the input image, we can neglect the global transformation between the GMMs. We define the following hypothesis test\(^{2}\) for the similarity between a mixture component of the input image and a mixture component of a given 2D aspect model

\[
H_0: (\mu_i^I, \Sigma_i^I) = (\mu_j^{P_j} + R_j \cdot T_j \cdot \Sigma_j^{P_j})
\]

\[
H_1: \quad \text{otherwise}
\]

reject $H_0$ if $TS_1 < \gamma_1$ or $TS_2 > \gamma_2$

where $\gamma_1$ and $\gamma_2$ are the thresholds and $TS_1$ and $TS_2$ are test statistics. $TS_1$ is the test statistic which defines the position on the object. Let

\[
v_i^I = \mu_i^I - oc_i \quad \text{and} \quad v_j^{P_i} = \mu_j^{P_i} - oc_j
\]

the vectors between mixture components and object center. Then $TS_1$ is given by

\[
TS_1 = \|v_i^I - v_j^{P_i}\|_2.
\]

Let $e_i^I$ be the principal eigenvector of $\Sigma_i^I$ and $e_j^{P_i}$ the principal eigenvector of $\Sigma_j^{P_i}$, then $TS_2$ is given by the scalar product between these eigenvectors

\[
TS_2 = e_i^I \cdot e_j^{P_i}.
\]

\(^{2}\)The statistical hypothesis, which has to be tested is the null hypothesis $H_0$. The alternative hypothesis is denoted by $H_1$. The test statistic $TS$ is a statistic on whose value the null hypothesis will be rejected or not. The threshold of rejecting a null hypothesis is given by $\gamma$ (notation is based on [11]).
The threshold $\gamma_1$ defines the allowed position difference, and $\gamma_2$ defines the allowed orientation difference. On the one hand, too low thresholds would not be able to handle shape deformations of different objects of a category. On the other hand, too high thresholds would not be able to handle different poses of an object. Our discrete similarity measure between two Gaussians is then given by

$$SM(f, g) = \begin{cases} 1 & \text{if } TS_1 < \gamma_1 \text{ or } TS_2 > \gamma_2 \\ 0 & \text{otherwise.} \end{cases}$$

(13)

To identify which pose hypothesis $P_p$ is the correct one we use a modification of the KL-divergence (6). Assuming that all mixture components have the same weight $\alpha_i = 1$ and $\beta_j = 1$ and using our own similarity measure $SM(f, g)$ instead of the KL-divergence $KL(f, g)$ between two mixture components we can rewrite (6) to

$$SM(f \parallel g) \approx \sum_{i=1}^{n} \alpha_i \min_j (SM(f_i \parallel g_j)).$$

(14)

Using this formulation, $SM(f \parallel g)$ defines the number of mixture components in $f$ which have a correct match in $g$. Similarly,

$$\hat{SM}(f \parallel g) \approx \sum_{i=1}^{n} \alpha_i \min_j (abs(SM(f_i \parallel g_j) - 1))$$

(15)

defines the number of mixture components in $f$ which have no correct match in $g$. For all 2D aspect models in $P$ the pose hypothesis $H(P)$ is given by

$$H(P) = \arg \max_{p \in P} \frac{SM(I \parallel P)}{SM(I \parallel \hat{P})}.$$  

(16)

3. Experiments and Results

We use 3D models (from [10], not trained on ETH80) of cows and horses, and build 3D car models from our own seven toy cars. Based on these 3D Gaussian Mixture contour models, we test our pose estimation algorithm on 17 poses of the ETH80 database [4] for the categories ‘horse’ and ‘cow’ and on some poses of the 3D object category dataset CAR [12]. We assume that the object has already been detected in a given input image, leaving us with the task of pose estimation.

3.1. 3D Gaussian Mixture Contour Models - 2D Aspect Models

As described in Section 2.1., we use a pose-invariant 3D contour category model based on GMMs for our pose estimation algorithm. Such a 3D Gaussian Mixture contour category model consists of a set of partitions of several training objects which are discriminative for them. Figure 2 shows an example of such a 3D model and two example 2D aspect models which were computed using the Unscented Transformation (see Section 2.2.). In our experiments, we choose several viewpoints over the top hemisphere around the object according to the tested database (see Figure 4). For purpose of visualization we show only discriminative mixture components of one training object.

We process a given 2D input image such that we first use a Canny edge detector and a linking algorithm to achieve long connected 2D contour fragments. Next, we compute a GMM for each contour fragment in the same way as described in Section 2.1. for 3D. Two example input images and their GMM representations are shown in Figure 3.

3In our experiments, typically we choose $\gamma_2 = 0.98$ and we choose $\gamma_1$ as 5%-10% of the bounding box size.
3.2. 2D Pose Estimation on the ETH80 database

In our experiments we estimate the pose of cows and horses of the ETH80 database. We select 17 poses of the ten horses and seven cows (except the lying cow and the cows with head-down) of the ETH80 database (see Figure 4 for a cow example of the selected 17 poses). We evaluate our 3D pose estimation algorithm by computing the similarity measure for the input pose to all cow or horse models of the 3D Gaussian Mixture Contour Model Database and compare these pose estimation votes with the ground truth given by the ETH80 database. The complexity of the pose estimation algorithm is linear in the number of poses. The reconstruction of 3D category models is computationally demanding, but needs to be done only once per category.

Figure 5(a) shows the confusion matrix of the pose estimation experiments on the ETH80 cows. We achieve an average accuracy of 68% of the pose estimation. We see that problems occur with
neighboring views (e.g. P14 votes not only for P14, but also for P8; P10 votes not only for P10, but also for P6 and P11) and pose ambiguity (e.g. P17 votes for P17 and P11). On the one hand, Figure 4 shows that neighboring views may look similar and so shape features have a similar appearance of position and orientation on the object. On the other hand, we see that pose ambiguity plays an important role. When working with 2D images we lose one important information: depth. In several poses e.g. front or back view of shape models (see Figure 1(d) and Figure 1(e) for an example) it is hard to differ on the basis of their shape. Apart from these two points, many poses can be estimated correctly, six poses achieve 100%.

Figure 5(b) shows the confusion matrix of the pose estimation experiments on the ETH80 horses. We achieve an average accuracy of 64% for the pose estimation. We see a similar behaviour as in the cow experiment for similar views (neighboring views, pose ambiguity). Figure 6 shows example pose estimation results on the ETH80 database.

![Confusion Matrix](image)

**Figure 5.** (a) Confusion matrix for pose estimation for seven cows of the ETH80 cows (average accuracy 68%); (B) Confusion matrix for pose estimation on the ETH80 horse dataset for all ten horses (average accuracy 64%).

![Pose Estimation Results](image)

**Figure 6.** Example pose estimation results of pose (a) P15 of an ETH80 cow, (b) P2 of an ETH80 cow, (c) P8 of an ETH80 horse, (d) P11 of an ETH80 horse.

In a further experiment we also test our pose estimation method on the segmented 3D object category dataset CAR ([12]). In the discrimination between front-back and left-right view we achieve an average accuracy of 94% which outperforms [1] and is comparable to [15] and [6] by considering pose ambiguities. But we do not need any training on this dataset in contrast to [15, 1, 6].

4. Conclusion

We have presented a pose estimation algorithm for 2D input images solely based on shape information. Our method uses a probabilistic 3D model based on Gaussian Mixture Models of 1D manifolds in 3D; 2D aspect models are then generated using an Unscented Transformation. We have introduced
a novel similarity measure between GMMs which is based on a hypothesis test. Our experiments on the ETH80 database show the possibility of pose estimation for the categories 'horse' (average accuracy 64%) and 'cow' (average accuracy 68%) for 17 views (standard eight views + nine views from above). Future work will include a combination of this pose estimation algorithm with object categorization. Moreover, such 3D Gaussian Mixture contour category models may also be useful for 2D object categorization on the basis of pose-invariant 3D models.

References


